Amendments to the Claims

1. (CURRENTLY AMENDED) A computer-based software task management system ~~(100),~~ comprising: an index register ~~(130)~~ configured to store a data register pointer for pointing to a data register ~~(150);~~ a Task ID register ~~(110)~~ coupled to the index register and configured to store a Task ID keyed to the index register; a Task ID memory ~~(120)~~ coupled to the Task ID register and configured to store a flag indicating whether the Task ID is available; and a state machine ~~(105)~~ coupled to the Task ID memory and configured (a) to receive a Task ID request from a task, (b) to determine whether a Task ID is available in response to the Task ID request, (c) when a Task ID is available, to issue a Task ID to the task and set the flag in the Task ID memory indicating that the Task ID is in use, and (d) when the task is complete, to reset the flag in the Task ID memory indicating that the Task ID is available.

2. (CURRENTLY AMENDED) The computer-based software task management system of claim 1, further comprising: a plurality of index registers ~~(130)~~ each configured to store a data register pointer for pointing to a data register ~~(150);~~ a plurality of Task ID registers ~~(110)~~ each coupled to the index register and each configured to store a Task ID keyed to a respective index register; a plurality of Task ID memories ~~(120)~~ each coupled to the Task ID register and configured to store a flag indicating whether a respective Task ID is available; and wherein the state machine ~~(105)~~ is configured to manage a plurality of tasks with the plurality of index registers, Task ID registers and Task ID memory.

3. (ORIGINAL) The computer-based software task management system of claim 2, wherein each index register is uniquely associated with a different Task ID.

4. (ORIGINAL) The computer-based software task management system of claim 2, further comprising: a flip-flop circuit coupled to the index register and configured to cause the task to alternate between a write cycle to the index register and one selected from the group consisting of: a write cycle to the data register pointed to by the index register; and a read cycle to the data register pointed to by the index register.

5. (CURRENTLY AMENDED) A method for managing multiple tasks using an index register ~~(130)~~, comprising: (a) receiving a Task ID request from a task; (b) determining whether a Task ID is available in response to the Task ID request; (c) when a Task ID is available, issuing a Task ID to the task and setting a flag in a Task ID memory ~~(120)~~

indicating that the Task ID is in use, and (d) when the task is complete, resetting the flag in the Task ID memory indicating that the Task ID is available.

6. (ORIGINAL) The method of claim 5 using a plurality of index registers with a Task ID associated with each index register, wherein: the determining step includes the step of determining whether a Task ID is available from the plurality of Task IDs in response to the Task ID request.

7. (ORIGINAL) The method of claim 5, further comprising: when a Task ID is not available, periodically requesting a Task ID.

8. (ORIGINAL) The method of claim 5, further comprising the step of: causing the task to alternate between a write cycle to the index register and one selected from the group consisting of: a write cycle to a data register pointed to by the index register; and a read cycle to the data register pointed to by the index register.

9. (ORIGINAL) The method of claim 6, further comprising the step of: causing the task to alternate between a write cycle to the index register and one selected from the group consisting of: a write cycle to a data register pointed to by the index register; and a read cycle to the data register pointed to by the index register.

10. (ORIGINAL) The method of claim 8, further comprising: resetting a flip-flop circuit after a read cycle, the flip-flop circuit steering the read and write accesses to the index register and the data register pointed to by the index register.

11. (ORIGINAL) The method of claim 9, further comprising: resetting a flip-flop circuit after a read cycle, the flip-flop circuit steering the read and write accesses to the index register and the data register pointed to by the index register.